



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
-----------------	-------------	----------------------	---------------------	------------------

10/553,636

10/19/2005

Wolfgang Flatow

052003-0014

6637

1923 7590 09/02/2008
MCDERMOTT, WILL & EMERY LLP
227 WEST MONROE STREET
SUITE 4400
CHICAGO, IL 60606-5096

EXAMINER

HWA, SHYUE JIUNN

ART UNIT

PAPER NUMBER

2163

MAIL DATE

DELIVERY MODE

09/02/2008

PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No. 10/553,636	Applicant(s) FLATOW, WOLFGANG	
	Examiner JAMES HWA	Art Unit 2163	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 12 June 2008.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-39 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-39 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
 2. ☐ Certified copies of the priority documents have been received in Application No. _____.
 3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|--|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413) |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | Paper No(s)/Mail Date. _____ |
| 3) <input checked="" type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08) | 5) <input type="checkbox"/> Notice of Informal Patent Application |
| Paper No(s)/Mail Date <u>6/12/2008</u> . | 6) <input type="checkbox"/> Other: _____ |

DETAILED ACTION

1. Applicant has amended claims 1 and 18 in the amendment filed on 06/12/2008. New claims 23-39 added. Claims 1-39 are pending in this Office Action.
2. A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on 06/12/2008 has been entered.

Response to Arguments

3. Applicant's arguments filed 06/12/2008 have been full considered but are not persuasive.

Examiner respectfully disagree all allegations as argued. Examiner, in her previous office action gave detail explanation of claimed limitation and pointed out exact locations in the cited prior art. Examiner is entitled to give claim limitations their broadest reasonable interpretation in light of the specification. See. MPEP 2111[R-1] Interpretation of Claims-Broadest Reasonable Interpretation During patent examination, the pending claims must be 'given the broadest reasonable interpretation consistent with the specification'. Applicant always has the opportunity to amend the claims during prosecution and broad interpretation by the examiner reduces the possibility that the claim, once issued, will be interpreted more broadly than is justified. In re Prater, 162 USPQ 541,550-51 (CCPA 1969).

Applicant argued that, the metadata tables of Lee are not a database schema in claim 1. Examiner respectfully disagrees.

In response Applicant's argument, Lee teaches it will be understood that the relational database can be any of the well-known relational databases. The first data storage portion is typically, and preferably, a set of tables in the relational database. The second data definition portion preferably comprises a relational schema as is typically used to model, outline or diagram the interrelationship between tables in a relational database. It is an important that the document-type definition (DTD) is loaded by the system and used in metadata format to generate the relational schema of the second data definition portion (column 15, lines 40-55; see also elements 14, 20, 22 of figure 1).

Applicant argued that, prior art Lee disclose the data representative of the DTD is stored in metadata tables, before being used by the generator to generate "not manage" a relational schema. Examiner respectfully disagrees.

Original instant specification and claim disclose "a database management system in accordance with a schema" is not "a database management schema".

Database management systems (DBMS) (disclose in the instant specification paragraph 0002 and Lee's specification column 4, line 65) is computer software designed for the purpose of managing databases based on a variety of data models.

In response Applicant's argument, Lee teaches relational schema definition is examined for XML data, a relational schema is created out of a DTD, and XML data is loaded into the generated relational schema that adheres to the DTD. In this manner,

Art Unit: 2163

the data semantics implied by the XML are maintained so that more accurate and efficient management of the data can be performed (column 5, lines 50-55).

Applicant argued that, Lee describes in column 18, lines 1 to 3 that the relational schema is achieved by mapping the metadata tables. This mapping requirement means that the schema would not have a similar structure to the metadata tables; otherwise such mapping would not be required. Examiner respectfully disagrees.

Original instant specification discloses mapping the tables in paragraph 0054 and 0139.

In response to applicant's argument that the references fail to show certain features of applicant's invention, it is noted that the features upon which applicant relies (i.e., "mapping the metadata tables", "This mapping requirement means that the schema would not have a similar structure to the metadata tables") are not recited in the rejected claim(s). Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993).

Applicant argued that, the schema of Lee does not satisfy the requirement of a second table related to the first table to store the names of entities (as in Lee each table stores only details of one entity and not plural entities), and there is also no table that stores names of entity types. Examiner respectfully disagrees.

In response Applicant's argument, Lee teaches the generator can alter the schema of the relational database to add a column to each table in the schema corresponding to each row of the metadata attribute table related to the particular

Art Unit: 2163

metadata item table row. The generator can alter the tables in the schema of the relational database to add links between tables in the schema corresponding to a relationship identified in each row of the metadata nesting table (column 8, line 46-54).

In the DTD, some attribute types can contain multiple values, e.g. IDREFS, NMTOKENS, ENTITIES. Such attributes can be analogized to a set rather than an attribute, and it is desired to represent their values as sets (column 28, lines 53-58). IBM has proposed the idea of breaking DTDs into elements, notations, and entities that use components grouped with properties sequential, choice, attribute, and relationship and with repetition properties to construct DTDs. Tools to do XML translation and XML generation from SQL queries are provided (column 5, lines 3-12).

Applicant argued that, the schema of Lee does not provide one or more value storage tables related to the second and third tables to associate stored field values with entities. Examiner respectfully disagrees.

In response Applicant's argument, Lee teaches when a link is encountered, three possible cases result. First, the foreign key in one table can be updated with the key value in another table. Second, if there is a group in this link, then a new tuple is created in the group table as well as the corresponding foreign keys are updated. Third, if the child node is inlined in the parent node, then all of the attributes of the child table are copied into the parent table (column 35, lines 32-40; see also elements 92, 98 of figure 1B and elements 76, 84 of figure 4).

Applicant argued in claim 1 that, the schema taught by Lee is a standard table-per-object schema in which each entity is defined by a respective table within the

Art Unit: 2163

database. Consequently, whenever it is needed to add a new entity, it is necessary to define a new table within the database.

Also, Applicant argued that, the claimed invention provides a generalized configuration that can replace existing "table-per-object" type schemas for all relational databases. This arrangement allows data, including new entities, to be added to the database without requiring the addition of new tables to an existing schema. Examiner respectfully disagrees.

In response to applicant's argument that the references fail to show certain features of applicant's invention, it is noted that the features upon which applicant relies (i.e., "it is needed to add a new entity, it is necessary to define a new table within the database", "table-per-object" and "a generalized configuration") are not recited in the rejected claim(s).

Since it appears that each and every element of the Applicant's claimed invention are either disclosed or suggested by Lee and O'Brien of the record, the rejections given in the preceding office action are sustained.

Specification

4. Specification is objected under 37 CFR 1.75 because the term "A computer-readable medium" in claim 1, is not defined in the specification. Appropriate correction is required.

Claim Objections

5. Claims 23-38 are objected to because of the following informalities: "the software product" should be written as "the software program" as claim. Appropriate correction is required.

Claim Rejections - 35 USC § 112

6. Claims 1 and 18 are rejected under 35 U.S.C. 112, second paragraph, as failing to set forth the subject matter which applicant(s) regard as their invention. Evidence that claims 1 and 18 fail(s) to correspond in scope with that which applicant(s) regard as the invention can be found in the original disclosure filed 10/19/2005. In that paper, applicant has stated "a database management system in accordance with a schema", and this statement indicates that the invention is different from what is defined in the claim(s) because "a database management system in accordance with a schema" is not "a database management schema". Appropriate correction is required.

Claim Rejections - 35 USC § 101

35 U.S.C. 101 reads as follows:

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefore, subject to the conditions and requirements of the title.

7. Claim 39 is rejected under 35 U.S.C.101 because the language of the claim raises a question as to whether the claim is directed merely to an abstract idea that is not tied to a technological art, environment or machine which would result in a practice

Art Unit: 2163

application producing a concrete, useful, and tangible result to form the basis of statutory subject matter under 35 U.S.C 101.

Claim 39 recite "A database schema for use in managing a database".

The claim 39 lack the necessary physical articles or objects to constitute a machine or a manufacture within the meaning of 35 USC 101. They are clearly not a series of steps or act to be a process nor are they a combination of chemical compounds to be a composition of matter. As such, they fail to fall within a statutory category. They are, at best, functional descriptive material *per se*.

Merely claiming nonfunctional descriptive material, i.e., abstract ideas, stored on a computer-readable medium, in a computer, or on an electromagnetic carrier signal, does not make it statutory. See *Diehr*, 450 U.S. at 185-86, 209 USPQ at 8 (noting that the claims for an algorithm in *Benson* were unpatentable as abstract ideas because "[t]he sole practical application of the algorithm was in connection with the programming of a general purpose computer.").

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein

Art Unit: 2163

were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later invention was made in order for the examiner to consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g) prior art under 35 U.S.C. 103(a).

8. Claims 1-14 and 16-39 are rejected under 35 U.S.C. 103(a) as being unpatentable over by Lee et al. (US Patent No. 7,031,956 B1) in view of O' Brien et al. (US Patent No. 6,470,343 B1, hereinafter "O' Brien").

As to claim 1, Lee teaches the claimed limitations:

"A computer software program recorded on a machine- readable medium and containing machine readable instructions for execution by an electronic processor to provide a database management system, the program comprising in accordance with a database" as a relational schema definition is examined for XML data, a relational schema is created out of a DTD, and XML data is loaded into the generated relational schema that adheres to the DTD. In this manner, the data semantics implied by the XML are maintained so that more accurate and efficient management of the data can be performed (column 5, lines 50-56). Lee also teaches the first data storage portion is typically, and preferably, a set of tables in the relational database. The second data definition portion preferably comprises a relational schema as is typically used to model, outline or diagram the interrelationship between tables in a relational database (column 15, lines 43-48).

Once validated, the execution processor modifies the relational tables of the relational database according to the specified update semantics (column 51, lines 18-20). The loader and data synchronizer comprises an XML repository management system which stores a RDBMS having metadata tables and generated relational tables 34 and 20, respectively (column 46, lines 63-67; see also elements 20, 22, 28 and 34 of figure 1).

A system for generating a relational schema from a document type definition, forming a relational database from the relational schema and loading the contents of an extensible document into the relational database according to the relational schema (column 12, lines 12-16).

“a database management schema comprising; A first table to store the names of various entity types” as a system is provided for generating a schema for a relational database corresponding to a document having a document-type definition and data complying with the document-type definition and loading the data into the relational database in a manner consistent with the relational schema (column 8, lines 2-8; see also element 30 of figure 1B). Relational database systems are a proven technology for managing business data and are used everywhere by various sizes of companies for their critical business tasks (column 3, lines 43-50).

“A second table related to the first table to store the names of entities of the various entity types” see elements 90, 96 of figure 1B.

“A third table related to the first table to store the names of fields in respect of the various entity types” as a relational database schema is generated from the metadata

Art Unit: 2163

tables including the step of forming tables for each element type in the metadata item table formed with default fields provided therein (column 12, lines 52-55; see also element 68 of figure 3 and element 92 of figure 1B).

“Identifiers to indicate the nature of the data to be stored in each of said tables” as the step of creating tables in the relational database (identified by reference number 52 in FIG. 2) (column 24, lines 30-32; see also elements 126, 128 of figure 6).

Lee does not explicitly teach the claimed limitation “one or more value storage tables related to the second and third tables to associate stored field values with entities”.

O’ Brien teaches a new attribute definition table configured to associate a new attribute with an existing table; and a new data table which stores new attribute values for core objects; thereby extending the core data model to allow the one or more core entities to be modified without affecting the customized extension to the core data model (claim 23). A user only has to add information to the same four tables to define any new entity and its owner, new attributes for new or core entities, and new data in the new attributes, simple data maintenance or input forms can be designed to manage extensions to the data model (column 4, lines 8-13).

Any number of Core attributes can be associated with any number of extended attributes, and so MEM relationship 207 has a many-to-one relationship with both Core Column 205 and MEM Column 203. As explained earlier, it will be seen that this aspect of the embodiment could be adapted to allow new relationships to be defined between new entities (column 4, lines 1-7).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made, having the teachings of Lee and O' Brien before him/her, to modify Lee one or more value storage tables related to the second and third tables because that would allow client and site-specific extensions necessitating little intervention from the software supplier and should leave the supplier free to update the core and make necessary changes to other client applications as taught by O' Brien (column 1, lines 58-63).

As to claim 2, Lee teaches the claimed limitations:

"A first hierarchical relationship applied to the first table and a second hierarchical relationship applied to the second table to facilitate definition of hierarchical entities" as store nesting relationships. After defining the PCDATA item and all group items, the hierarchical definitions of elements can be described as nesting relationships between two items. An element definition is a sequence of n sub-elements stored as n nesting relationships with index fields in the DTDM-Nesting table (column 22, lines 41-46; see also figure 1B, dash lines between elements 90, 96 and 94,100).

As to claim 3, Lee teaches the claimed limitations:

"The schema includes tables to store relationships between the entities" as altering the tables in the schema of the relational database to add links between tables in the schema corresponding to a relationship identified in each row of the metadata nesting table (column 7, lines 26-29).

As to claim 4, Lee teaches the claimed limitations:

“The first table includes a column to store pointers corresponding to entity types the pointers indicating locations from which default values may be obtained during creation of new instances of the entity types” as the method can comprise the step of calling both the create element primitive and the move element primitive to create a new element in the document object and to attach the newly-created element within the document object at a desired location (column 10, lines 5-9).

The method can further comprise the step of determining whether a default value is required during the creation of a new element in accordance with the at least one proposed data update during the step of validating the at least one proposed data update (column 7, lines 55-59).

As to claim 5, Lee teaches the claimed limitations:

“The third table includes a column to store data indicating that a newly created entity's name is to be generated from data stored in columns of the one or more value storage tables” as document storing is accomplished in two ways. First, the XML document is stored as a whole for indexing, referred to as way of storing the XML document as an XML column. Second, pieces of XML data are stored into table(s), referred to as XML collections (column 5, lines 14-18).

generating the schema for the relational database from the metadata, wherein at least one table is thereby defined in the relational database corresponding to at least

Art Unit: 2163

one content particle of the document-type definition via the metadata, and at least one column is defined in each of the at least one table corresponding to another of at least one content particle of the document-type definition (column 6, lines 46-53).

As to claim 6, Lee teaches the claimed limitations:

“The one or more value storage tables comprise a number of value tables each including a column of values of a particular type” as generating an attribute metadata table corresponding to attribute type content particles in the document-type definition; creating a default attribute value in the attribute metadata table corresponding to any default items in the item metadata table (column 6, lines 63-67).

The path can comprise at least one node indicator comprising a label value and a position value corresponding to the document object. The position value can correspond to a lateral sibling location in the document object. The label value can be selected based upon a node type of a predetermined node (column 10, lines 17-23). The method can further comprise the step of determining whether an enumerated value is required by the at least one proposed data update and determining whether a proposed value therefore is contained in a specified enumeration during the step of validating the at least one proposed data update (column 10, lines 62-67).

As to claim 7, Lee teaches the claimed limitations:

“One or more of the value tables are each related to one or more other tables of the schema” as every node preferably has a type and possibly one or more attributes.

Art Unit: 2163

Every attribute preferably has a name and its corresponding value. For internal nodes, the element type is written above the node followed by any attributes and their values. All leaf nodes have type PCDATA, and have their values in the value attribute below the node (column 33, lines 20-28).

Decomposes multiple tuples for a multi-value attribute and stores those values into column in table. For example, if a multi-value attribute has value v1 v2, then it will create two tuples with value v1 and v2 respectively (column 35, lines 15-19). When an attribute is encountered, two possible cases result, first, this attribute can be mapped into a column and then the column of the tuple in the specific table is updated. Second, this attribute can be mapped into a table, and then multiple tuples in the specific table can be created for each value in this attribute (column 35, lines 26-31).

As to claim 8, Lee teaches the claimed limitations:

“The one or more of the value tables are each related to the second table” as the optimizer, to create the link pattern and pattern mapping tables. It should be understood that the optimizer is entirely optional and can be omitted without departing from the scope of this invention. The tables comprise an IM-Item table, which contains mapping information relating to the DTDM-Item table, an IM-Attribute table that contains mapping information relating to the DTDM-Attribute table (column 16, lines 34-42; see also figure 1A).

As to claim 9, Lee teaches the claimed limitations:

“The one or more of the value tables are arranged to store pointers to data stored external to data structures created by the computer software product.” as the method comprising the steps of receiving at least one proposed data update representative of the supplemental data from a source external to the relational database (column 9, lines 39-41). The path can comprise at least one node indicator comprising a label value and a position value corresponding to the document object. The position value can correspond to a lateral sibling location in the document object (column 10, lines 17-22; see also element 12 of figure 16).

As to claim 10, Lee teaches the claimed limitations:

“The schema includes a data type table relating names of the value storage tables to corresponding names of the column of values of a particular type” as the relational database having a set of tables defined by a relational schema, the supplemental data comprising formatted data having a document type definition representative of the relational schema and represented in a document object (column 10, lines 28-32). Attribute-list declarations define attributes of an element type. The declaration includes attribute names, default values and types (column 2, lines 4-6).

As to claim 11, Lee teaches the claimed limitations:

“The data type table is related to the third table” as a DTDM-Attribute table generally made up of Attribute of elements and groups contained in the DT (column 16, lines 28-29; see also elements 90 and 92 in figure 1B).

As to claim 12, Lee teaches the claimed limitations:

“The data type table is related to an intermediate value type table and wherein the value type table points to the third table” as processing moves to a query of the DTDM-Item table is performed to return all of the item types stored in the DTDM-Item table. A proposed SQL statement to accomplish this task is shown in note 122 associated with step 120 in FIG. 6 (column 24, lines 30-33; see also element 122 of figure 6 and elements 90, 92 of figure 1B).

As to claim 13, Lee teaches the claimed limitations:

“The third table includes columns to define multiple field functionality” as processing then moves to step in which all of the nesting relationships from this group to its children are created by function fill_DTDM-Nesting_Item. Processing then moves to decision block in which it is determined whether there are additional element types in the DTD to be processed for the loop initiated (column 20, lines 14-22; see also element 424 of figure 5 and element 444 of figure 5A).

As to claim 14, Lee teaches the claimed limitations:

“The third table includes a column to indicate if historical data values are to be stored in respect of a corresponding field type and wherein the value storage tables each include a column to store current values of said field type and to store data indicating when the current values were written” as processing moves to step 120 in

Art Unit: 2163

which a query of the DTDM-Item table 90 is performed to return all of the item types stored in the DTDM-Item 90 table. A proposed SQL statement to accomplish this task is shown in note 122 associated with step 120 in FIG. 6. Processing then moves to step 124, which initiates a loop for every item returned in the record set selected in step 120. Processing within the loop then moves to step 126 wherein a table 20 is created in the relational database 14 with some key-type default fields, wherein the table name created in the database 14 corresponds to the Name field in the DTDM-Item table 90 as returned in the record set in step 120 (column 24, lines 33-44; see also figure 6).

As to claim 16, Lee teaches the claimed limitations:

“The schema includes a format table having columns to store data storage formats” as a system for synchronizing and updating a relational database containing existing data with supplemental data, the relational database having a set of tables defined by a relational schema, the supplemental data comprising formatted data having a document type definition representative of the relational schema and represented in a document object (column 10, lines 26-32). It is an important feature that the DTD is loaded by the system and used in metadata format to generate the relational schema of the second data definition portion (column 15, lines 49-55).

As to claim 17, Lee teaches the claimed limitations:

“The schema includes one or more tables to store values indicating groupings of sets of fields” as an element type definition, elements that are associated within

Art Unit: 2163

parentheses participate in a grouping relationship, and are defined as a group (column 2, lines 63-65). The loading step can further comprise the steps of: initializing a link table; determining whether each item in the metadata nesting table contains a group type; initializing a pattern-mapping table; directly mapping a link into the link table for each item in the metadata nesting table that does not contain a group type; creating an additional link table containing a mapping of a link pattern for each group type identified in the metadata item table (column 7, lines 37-45).

As to claim 18, Lee teaches the claimed limitations:

“A method implemented by means of an electronic processor to store data, the data concerning a number of entities of various entity types and relationships between the various entity types, the method comprising the steps of: providing the database management schema for storing identifiers of each of the entity types in a first table of the schema” as a method for generating a schema for a relational database corresponding to a document having a document-type definition and data complying with the document-type definition (column 6, lines 30-40). A DTD is used to define allowable structures of elements in a valid XML document. A DTD can basically include four kinds of declarations: element types attribute lists, notations, and entity declarations (column 1, lines 55-60). The loader and data synchronizer comprises an XML repository management system which stores a relational database management system (RDBMS) having metadata tables and generated relational tables (column 47, lines 61-67; see also element 14, 20, 34 of figure 16).

A location of a particular element in the document object can be identified by a path. The path can be formed in an XPath syntax. The method can also further comprise the step of converting an object identifier of a particular document object into a path (column 10, lines 4-10).

“storing identifiers of each of the number of entities in a second table of the schema, the second table being related to the first table; storing identifiers of each of a number of field types for the various entity types in a third table of the schema, the third table being related to the first table; and storing field values associated with the entities in one or more value storage tables of the schema, the one or more value storage tables being related to the second and third tables” as the DOM operations are object-oriented operations, in the sense that every node is identified by its Object Identifier (OID). However, we cannot guarantee that the OID used by the external XML data source will be the same as the internal IID used in the relational storage, because they are generated by two separate systems. To assure identification of the matching elements across these two systems, the identification of items in DOM could be achieved by assuming that the XML data source uses an XPath expression to uniquely identify the to-be-modified node (column 55, lines 26-37; see also OID between elements 122, 123 of figure 24). Entities, types and values see table PCDATA of figure 24.

As to claims 19-20, the limitations therein have substantially the same scope as claims 2-3. In addition, Lee teaches a system and method for receiving XML-based data

Art Unit: 2163

and updating a set of relational database tables only with data that has changed and verifying that the updates have performed successfully (column 1, lines 22-26).

Document object model (DOM) operations are object-oriented operations, in the sense that every node is identified by its Object Identifier (column 55, lines 27-28). In the loader and data synchronizer, elements, when mapped to their relational representation, is identified by the pair node-name and iid (column 57, lines 35-37).

These claims are rejected for at least the same reasons as claims 2-3.

As to claim 21, Lee teaches the claimed limitations:

“The step of storing data defining relationships includes: storing data identifying various relationship types in a fifth table; and storing data identifying relations in a sixth table” as in order to accomplish these functions, the system comprises an extractor, an optimizer, a generator and a loader all of which are interconnected to a storage unit. As contemplated by this, the storage unit comprises at least a metadata table storage portion and a pattern mapping table storage portion (column 15, lines 56-61; see also elements 36, three of them in figure 1B).

As to claim 22, Lee teaches the claimed limitations:

“A computational device operated according to the method of claim 18” as an execution device is operably connected to the translator for propagating the received at least one proposed data update into the relational database in a manner which ensures

Art Unit: 2163

compliance with both the relational database relational schema and the document type definition (column 10, lines 35-40).

As to claim 23, Lee teaches the claimed limitations:

“A computer software product according to claim 1, wherein the electronic processor manages the database in accordance with the database schema” as to import XML documents into a relational database management system (RDBMS), one step is to identify a relational schema for the XML documents. It is assumed that each document conforms to a DTD (column 48, lines 53-60).

As to claim 24, Lee teaches the claimed limitations:

“stores the database schema; and, manages the database using the stored schema” as the DTD Loader and Dumper loads a DTD, provided by an external source, into the RDBMS, and stores it as metadata in what is called the DTDM tables (column 48, lines 16-23).

As to claim 25, Lee teaches the claimed limitations:

“accesses the schema; and, interacts with the database using the schema” as in order to be able to make use of the characterized metadata and constraints expressed in DTDs, the DTD information is stored into relational tables as well as the XML data. This provides a uniform interface for tools such as the importer and synchronizer to

Art Unit: 2163

access both data and metadata, namely, via an appropriate query language, such as SQL (column 49, lines 22-32).

As to claim 26, Lee teaches the claimed limitations:

“ wherein the interaction includes at least one of: a) viewing data; b) editing data; c) listing data; d) searching data; and, a) reporting” as a structural view can be chosen to avoid having to choose particular semantics for all such relationships. For example, line one of the DTD in Example 1 specifies that in a book, a booktitle precedes a list of authors (or an editor) (column 3, lines 3-12).

As to claim 27, Lee teaches the claimed limitations:

“wherein the electronic processor is for converting a first database to a second database, by, for each first database table: a) storing a name of the first database table as an entity type in a first table in the second database; b) for each field of the first database table, storing a field name as a field type in a third table related to the first table; c) for each entry in the first database table: i) storing an entity representing the entry in a second table related to the first table; and ii) storing values for each of a number of fields in the entry, in at least one value storage table related to the second and third tables” as the PCDATA item will be used to convert all element-to-PCDATA relationships, such as found in a mixed content definition, to element-to-element relationships, thus unifying these two types of nesting relationships. A PCDATA item

Art Unit: 2163

has one attribute called value that is used to capture the text value of this PCDATA item (column 22, lines 16-23; see also elements 90, 92, 94 of figure 1B).

As to claim 28, Lee teaches the claimed limitations:

“Wherein the electronic processor is for: a) storing an entity type identifier for each entity type in the first table; b) for each entity in the second table, storing an entity identifier together with the entity type identifier of the corresponding entity type; c) for each field type in the third table, storing a field type identifier together with the entity type identifier of the corresponding entity type; d) for each field in the value table, storing: i) a field identifier; ii) a corresponding value for the field; iii) a corresponding entity identifier for the field; and, iv) a corresponding field type identifier for the field” as the metadata tables are fed to the generator and, optionally, the optimizer, to create the link pattern and pattern mapping tables. The tables comprise an IM-Item table which contains mapping information relating to the DTDM-Item table, an IM-Attribute table that contains mapping information relating to the DTDM-Attribute table, an IM-Nesting table that contains mapping information relating to the DTDM-Nesting table, and a TS-JC table for containing table schema and join constraint information for assisting in the generation of the table schema for the relational database (column 16, lines 32-55; see also figure 1B).

As to claim 29, Lee teaches the claimed limitations:

Art Unit: 2163

“wherein the second table includes, for each stored entity, an indication of a corresponding entity type stored in the first table” as it has been assumed that entity declarations can be substituted or expanded to give an equivalent DTD with only element type and attribute-list declaration (column 2, lines 12-18).

As to claim 30, Lee teaches the claimed limitations:

“Wherein the third table includes, for each stored field type, an indication of a corresponding entity type stored in the first table” as (element 92, 98 of figure 1B).

As to claim 31, Lee teaches the claimed limitations:

“Wherein the schema includes a fourth table related to the second and third tables for storing fields and wherein, at least one of: the one or more value tables are related to the fourth table; and, the one or more value tables are the fourth table” as (elements 92, 98, 102 of figure 1B).

As to claim 32, Lee teaches the claimed limitations:

“Wherein the one or more storage value tables include, for each stored field value:

a) an indication of a corresponding entity stored in the second table; and, b) at least one of: iii) an indication of a corresponding field type stored in the third table; and, iv) an indication of a corresponding field stored in a fourth table” as (elements 90, 96, 36 of figure 1B).

As to claim 33, Lee teaches the claimed limitations:

“Wherein the first table includes at least: an identifier column including an identifier for each entity type; and, a name column including a name for each entity type” as the DOM operations are object-oriented operations, in the sense that every node is identified by its Object Identifier (OID) (column 55, lines 26-30 ; see also figure 15).

As to claim 34, Lee teaches the claimed limitations:

“wherein the second table includes at least: a) an identifier column including an identifier for each entity; b) a name column including a name for each entity; and, c) an entity type identifier column including an identifier for a corresponding entity type for each entity” as DTDM_item table (elements 90, 96 of figure 1B).

As to claim 35, Lee teaches the claimed limitations:

“wherein the third table includes at least: a) an identifier column including an identifier for each field type; and, b) a name column including a name for each field type; and, c) an entity type identifier column including an identifier for a corresponding entity type for each field type” as DTDM_attribute table (elements 92,98 of figure 1B).

As to claim 36, Lee teaches the claimed limitations:

“Wherein the at least one value table includes at least: an identifier column including an identifier for each value; and, a value column including a value for each field” as itemized call table (elements 116-118 of figure 24).

As to claim 37, Lee teaches the claimed limitations:

“wherein the at least one value table includes: an entity identifier column including an entity identifier indicative of a corresponding entity for each field; and, an field type identifier column including a field type identifier indicative of a corresponding field type for each field” as itemized call table (elements 116-118 of figure 24).

As to claim 38, Lee teaches the claimed limitations:

“Wherein the one or more value tables include a field identifier column including an identifier for a corresponding field for each value, and wherein the schema includes a fourth table, the fourth table including. an entity identifier column including an entity identifier indicative of a corresponding entity for each field; and, an field type identifier column including a field type identifier indicative of a corresponding field type for each field” as PCDATA table (figures 17 and 24).

As to claim 39, Lee teaches the claimed limitations:

“A database schema for use in managing a database, the database schema” as way to import XML data into a relational database schema. Current industry enterprise database management system (DBMS) vendors, such as DB2 and Oracle 8i, provide

Art Unit: 2163

XML extensions for bringing XML data into a relational database (column 4, lines 3-15).

A relational database schema is generated from the metadata tables including the step of forming tables for each element type in the metadata item table (column 12, lines 50-55).

“a) a first table to store the names of various entity types” as (element 30 of figure 1B).

“b) a second table related to the first table to store the names of entities of the various entity types” as (element 36 of figure 1B).

“c) a third table related to the first table to store the names of fields in respect of the various entity types” as (elements 90, 96 of figure 1B).

“e) identifiers to indicate the nature of the data to be stored in each of said tables” as a flow chart detailing the steps of converting a DOM tree object identifier into an XPath expression for use in updating an RDBMS (column 13, lines 57-60). DOM Object Identifier (OID) to XPath Expression. We can see that the DOM operations are object-oriented operations, in the sense that every node is identified by its OID (column 55, lines 26-33; see OID between elements 122 and 123 of figure 25).

Lee does not explicitly teach the claimed limitation “one or more value storage tables related to the second and third tables to associate stored field values with entities”.

O’ Brien teaches a new attribute definition table configured to associate a new attribute with an existing table; and a new data table which stores new attribute values for core objects; thereby extending the core data model to allow the one or more core entities to be modified without affecting the customized extension to the core data

Art Unit: 2163

model (claim 23). A user only has to add information to the same four tables to define any new entity and its owner, new attributes for new or core entities, and new data in the new attributes, simple data maintenance or input forms can be designed to manage extensions to the data model (column 4, lines 8-13).

Any number of Core attributes can be associated with any number of extended attributes, and so MEM relationship 207 has a many-to-one relationship with both Core Column 205 and MEM Column 203. As explained earlier, it will be seen that this aspect of the embodiment could be adapted to allow new relationships to be defined between new entities (column 4, lines 1-7).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made, having the teachings of Lee and O' Brien before him/her, to modify Lee one or more value storage tables related to the second and third tables because that would allow client and site-specific extensions necessitating little intervention from the software supplier and should leave the supplier free to update the core and make necessary changes to other client applications as taught by O' Brien (column 1, lines 58-63).

9. Claim 15 is rejected under 35 U.S.C. 103(a) as being unpatentable over Lee et al. (US Patent No. 7,031,956 B1) and O' Brien et al. (US Patent No. 6,470,343 B1) as applied to claim 1 above, and further in view of Iborra et al. (US Patent Application No. 2003/0167455 A1, hereinafter "Iborra").

As to claim 15, Lee does not explicitly teach the claimed limitation “the third table includes a column to store values indicating whether or not values of a newly created instance of an entity are to be inherited from another instance of an entity”.

Iborra teaches the model also maintains information on relationships between classes, which can be of two types: aggregation and inheritance. Each inheritance relationship stores the name of the parent class, the name of the child class and whether the specialization is temporary or permanent. Finally, if the specialization is permanent it stores a well-formed formula on constant attributes as specialization condition (page 8, paragraph 0095). The system logic translator is a translator that writes the code that actually carries out the processing of all the services defined in the objects defined by the Conceptual Model to alter the values of attributes of various objects, call services of other objects (page 3, paragraph 0035).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made, having the teachings of Lee, O’ Brien and Iborra before him/her, to modify Lee created instance of an entity are to be inherited from another instance of an entity because that would allow a user to input the requirements, a validator for validating the input requirements as taught by Iborra (page 7, paragraph 0083).

Contact Information

Any inquiry concerning this communication or earlier communications from the examiner should be directed to James Hwa whose telephone number is 571-270-1285.

Art Unit: 2163

The examiner can normally be reached on 8:00 – 5:00. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Don Wong can be reached on 571-272-1834. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR.

Status information for unpublished applications is available through Private PAIR only, for more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the PAIR system contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

08/23/2008

/James Hwa/
Examiner, Art Unit 2163

/Cam Y Truong/
Primary Examiner, Art Unit 2169